



Ordenación #1

Transcripción

[00:00] Bien, ya tenemos la funcionalidad básica para paginar. Para fines didácticos yo he creado unos registros más aquí en nuestro registro de médicos, para de esta forma ver el efecto de cómo podemos ordenarlo.

[00:17] Yo he agregado algunos nombres más con otras letras diferentes, porque lo que yo quiero ahora, siguiendo las reglas del negocio, es ordenar, ya sea por documento o por nombre. Similar a la paginación, ordenación es también un caso de uso súper súper común y Spring también nos facilita la vida con ello.

[00:39] Ustedes se preguntarán, ¿cómo? Simple. Agregamos un ampersand más a los query params o parámetros de query y le decimos sort. Sort significa orden en inglés. Si se dan cuenta, todos los atributos que estamos usando aquí del query params, que son los atributos de pageable, son en inglés.

[00:58] De cierta forma, es muy bueno que se vayan familiarizando con que todo sea en inglés, incluso nombrar los atributos en inglés. Es como la documentación también la van a encontrar en todas partes. No nos vamos a a separar del tema, vamos a decirle que ahora lo queremos ordenado por nombre.

[01:14] Le vamos a dar enviar y vemos que en efecto ahora está ordenado ascendentemente por nombre, vemos aquí a Ángela, Betty y Diego, dos veces repetido Erick y Oscar alfabéticamente por nombre. Si lo quiero por

documento, entro aquí, envío y vemos que él cambia el orden porque ya está por número de documento.

[01:38] Ahora, una cosa, un detalle que tienen que ver aquí. El atributo que estamos poniendo aquí debe ser el atributo como está registrado en nuestra entidad de la base de datos. ¿Qué quiere decir esto? Por ejemplo si yo en mi DTO vamos a suponer en mi DTO documento, yo lo he llamado código.

[02:06] Yo le puse código y si yo aquí registro sort por código, entonces no va a funcionar porque internamente, ya que Spring está haciendo la paginación, Spring va a buscar por ese atributo código dentro de mi entidad médico y no lo va a encontrar porque yo acá tengo documento, entonces dato muy pero muy importante. El atributo de orden debe ser el mismo que tenemos en la entidad de la base de datos.

[02:37] Eso es lo primero. Segundo, y esto es un dato importante y también puede ser un artificio, por así decirlo, porque supongamos que yo tengo un orden por defecto que yo quiero seguir y de repente ustedes me dicen: “Diego, si yo tengo un orden por default, ¿siempre tengo que enviar estos parámetros?”

[03:03] Bueno, vamos por partes, primero estos parámetros, como ya lo vimos, no son obligatorios. Estos son parámetros, digamos para arreglar la lista que estamos recibiendo de vuelta de nuestro cliente.

[03:15] Por ejemplo, si yo le quito todos los parámetros y envío esto, yo voy a recibir la lista entera y con los valores por defecto que, por ejemplo, la lista del valor por defecto es 20 y todas esas cosas. ¿Pero qué pasa si yo le digo no, pero yo quiero que por defecto no sea 20, por defecto, yo quiero que sea 10?

[03:36] ¿Existirá forma de modificar esto o es que es Spring ya lo tiene todo cerrado y es imposible? Bueno, como le dije antes, Spring es muy flexible y sí, sí podemos modificar estos valores por defecto. Lo único que tenemos que hacer es volver a nuestro controller y trabajar a nivel de esta interfaz, pageable

con una anotación que es `PageableDefault`, entonces le vamos a poner aquí `PageableDefault`.

[04:06] Y aquí en `PageableDefault` vamos a abrir paréntesis. Y vemos aquí una serie de parámetros que nosotros podemos agregarle aquí a esta anotación. Por ejemplo, `values`, `size`, `page`, `sort`, `direction`. Por ejemplo, vamos a decirle `size`.

[04:27] Vamos a decir `size` y queremos que el tamaño por defecto nos retorna solamente 2. Quiero solo 2 elementos en mi lista por defecto en este `PageableDefault`. El servidor ya recargó, esperamos un poco. Está listo y vamos a ver si es verdad, porque yo no le estoy enviando ningún parámetro. Le envío ahora.

[04:48] Y, en efecto, miren ahora, no he enviado ningún solo parámetro. Recibo solamente los dos primeros elementos que tengo en la base de datos. Y aquí, en los detalles del `pageable`, ya veo que el `size` ahora es 2. ¿Por qué? Porque `PageableDefault` sobrescribe, perdón, `PageableDefault` ya me dice cuáles son los valores por defecto y sobrescribe los que Spring ya tiene por dentro.

[05:17] Y si por ejemplo yo ahora hago el signo de interrogación para comenzar con los `query params`, y aquí hago `size`, el mismo que yo estoy estableciendo aquí, le digo: “No, yo, ahora quiero que me retornes 4”. Le envío y retorna 4. ¿Por qué? Porque ahora lo que yo envío, lo que mi cliente manda sobrescribe lo que `PageableDefault` tiene.

[05:46] Entonces para resumir esto, `pageable` por sí solo tiene sus valores por defecto, que ya vimos que es 20, inicia en la página 0, etcétera, lo pueden ver más a detalle aquí, si yo quiero sobrescribir los valores por defecto en `pageable`, uso `pageableDefault`.

[06:04] Con `pageableDefault` yo le especifico los nuevos valores por defecto que va a tener. Y si yo especifico manualmente los valores a nivel de mi URL en mis `query params` vamos a ver que eso sobrescribe lo que yo puse en

pageableDefault. ¿Se encuentra la jerarquía? Como siempre les digo, todo es cuestión de que sigan practicando.

[06:30] Queda un par de temas más por ver, que son actualización y delete de lo que es nuestro API. Nos vemos en la siguiente clase.